

ДОДАТОК А

Структуровані дані системи (Rule Cards та Prompt Contracts)

У додатку наведено рисунки А.1 та А.2, як приклади структурованих даних системи: Rule Cards (YAML) та Prompt Contracts (JSON).

```

1  card_id: CABAP-DB-001
2  title: Avoid SELECT in LOOP (N+1)
3  source: Clean ABAP + team rules
4  tags: [performance, db]
5  atc_hints: [CI_SEL_NESTED, "SELECT SINGLE in LOOP", "nested SELECT"]
6  category: Performance
7  problem: >
8      Database access inside a loop causes N+1 round-trips and unstable runtime
9      under load.
10 recommendation: >
11     Move database read out of the loop: collect keys, perform a bulk SELECT (or
12     JOIN/FAE),
13     build a hashed/sorted table, and read from memory inside the loop.
14 examples:
15     bad: |
16         LOOP AT lt_items INTO DATA(ls_item).
17             SELECT SINGLE * FROM zt_item INTO @DATA(ls_db) WHERE id = @ls_item-id.
18         ENDLOOP.
19     good: |
20         SELECT * FROM zt_item INTO TABLE @DATA(lt_db) FOR ALL ENTRIES IN @lt_items
21         WHERE id = @lt_items-id.
22         DATA lt_db_h TYPE HASHED TABLE OF zt_item WITH UNIQUE KEY id.
23         lt_db_h = CORRESPONDING #( lt_db ).
24         LOOP AT lt_items INTO DATA(ls_item).
25             READ TABLE lt_db_h INTO DATA(ls_db) WITH KEY id = ls_item-id.
26         ENDLOOP.
27 validation:
28     - Rerun ATC for affected objects
29     - Run ABAP Unit tests
30     - Optional: SAT/ST05 runtime comparison
31 constraints:
32     - Do not change business semantics
33     - Respect clean-core / released API restrictions
34 risk_level: medium

```

Рисунок А.1 – Приклад Rule Card (YAML формат)



```
1  {
2    "finding": {
3      "check_id": "CI_SEL_NESTED",
4      "category": "Performance",
5      "priority": 2,
6      "message": "SELECT SINGLE in LOOP"
7    },
8    "code_context": {
9      "snippet": "LOOP AT lt_items... SELECT SINGLE ... ENDLOOP.",
10     "signals": [
11       "has_select_in_loop",
12       "uses_select_single"
13     ]
14   },
15   "retrieved_rules": [
16     {"card_id": "CABAP-DB-001", "title": "Avoid SELECT in LOOP (N+1)"}
17   ],
18   "constraints": {
19     "no_semantic_change": true,
20     "clean_core": true,
21     "require_validation": true
22   }
23 }
24
25
```

Рисунок А.2 – Пример Prompt-объекта (JSON)

ДОДАТОК Б

Вихідний код прототипу (Python)

У додатку наведено рисунок Б.1, повний вихідний код прототипу рекомендаційної системи, реалізованого на мові Python. Код демонструє логіку роботи pipeline: отримання Finding, пошук правил у Knowledge Base (емуляція Retrieval) та генерація структурованої рекомендації. Даний прототип використовувався для експериментального дослідження ефективності системи.

```

abap_advisor.py x +
1 import json
2
3 # --- 1. БАЗА ЗНАНЬ (KNOWLEDGE BASE) ---
4 # Емуляція Rule Cards з Додатку А
5 KNOWLEDGE_BASE = [
6     {
7         "id": "CABAP-DB-001",
8         "category": "Performance",
9         "keywords": ["SELECT", "LOOP", "NESTED", "INTERNAL TABLE"],
10        "title": "Avoid SELECT in LOOP (N+1)",
11        "description": "Database access inside a loop causes N+1 round-trips.",
12        "recommendation": "Move database read out of the loop: collect keys, perform a bulk SELECT.",
13        "checklist": ["Rerun ATC", "Run ABAP Unit", "Check SAT/ST05"]
14    },
15    {
16        "id": "CABAP-ITAB-001",
17        "category": "Clean ABAP",
18        "keywords": ["DEFAULT KEY", "TABLE", "SORT"],
19        "title": "Use the right table type",
20        "description": "DEFAULT KEY can lead to unexpected sorting results.",
21        "recommendation": "Explicitly define the sort key or use HASHED/SORTED tables.",
22        "checklist": ["Replace DEFAULT KEY", "Use HASHED TABLE for lookups"]
23    }
24 ]
25
26 def get_sample_finding():
27     """Емуляція отримання Finding від ATC."""
28     return {
29         "check_id": "CI_SEL_NESTED",
30         "message": "SELECT SINGLE in LOOP detected",
31         "code_snippet": "LOOP AT lt_orders. SELECT SINGLE * FROM zcustomers ... ENDLOOP.",
32         "priority": "High"
33     }
34
35 def retrieve_rules(finding_text, kb):
36     """Етан Retrieval: Пошук релевантних правил."""
37     relevant_rules = []
38     text_lower = finding_text.lower()
39     for rule in kb:
40         # Простий алгоритм співпадіння за ключовими словами
41         match_score = sum(1 for kw in rule["keywords"] if kw.lower() in text_lower)
42         if match_score > 0:
43             relevant_rules.append((rule, match_score))
44     # Сортування за релевантністю
45     relevant_rules.sort(key=lambda x: x[1], reverse=True)
46     return [item[0] for item in relevant_rules]
47
48 def generate_recommendation(finding, rules):
49     """Етан Generation: Формування JSON-відповіді (симуляція LLM)."""
50     if not rules:
51         return {"status": "NO_RULE", "message": "No relevant rule found."}
52     best_rule = rules[0]
53     return {
54         "status": "SUCCESS",
55         "summary": f"Found issue: {finding['message']}.",
56         "rule_applied": best_rule["title"],
57         "clean_abap_alignment": best_rule["id"],
58         "suggested_change": best_rule["recommendation"],
59         "validation": best_rule["checklist"],
60         "risk_level": "medium"
61     }
62
63 def main():
64     print(">>> ABAP CODE QUALITY ADVISOR (PROTOTYPE) <<<")
65     print("-" * 50)
66     # 1. Отримання даних
67     finding = get_sample_finding()
68     print(f"[INPUT] ATC Finding: {finding['check_id']}")
69     # 2. Retrieval
70     found_rules = retrieve_rules(finding["message"], KNOWLEDGE_BASE)
71     print(f"[RETRIEVAL] Found {len(found_rules)} relevant rules.")
72     # 3. Generation
73     rec = generate_recommendation(finding, found_rules)
74     # 4. Output (UI Simulation)
75     if rec["status"] == "SUCCESS":
76         print("\nRECOMMENDATION REPORT ---")
77         print(f"Rule: {rec['rule_applied']}")
78     if __name__ == "__main__":
79         main()
80

```

Рисунок Б.1 – Основний модуль системи (abap_advisor.py)

ДОДАТОК В

Макети екранних форм прототипу

У додатку наведено концептуальні макети на рисунках В.1 та В.2 екранних форм прототипу рекомендаційної системи, що демонструють інтеграцію у робочий процес розробника: список findings, вибір порушення, retrieved rule cards, summary, recommended actions i validation checklist.

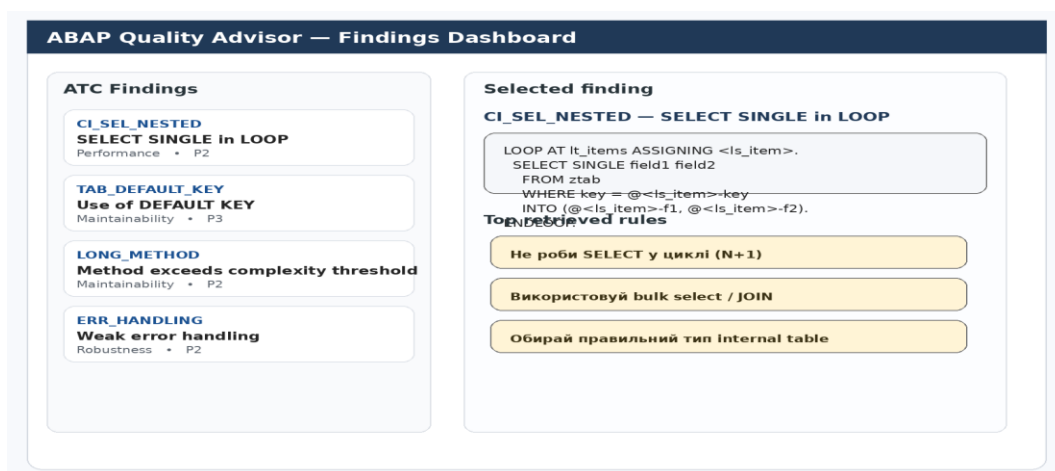


Рисунок В.1 – Макет екрана списку findings та retrieval-підказок

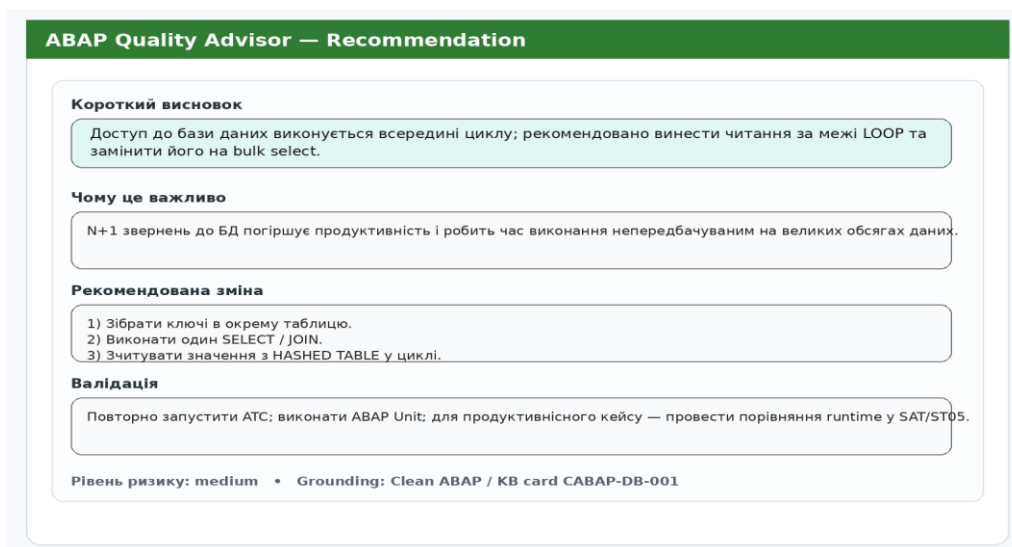


Рисунок В.2 – Макет екрана детальної рекомендації та керування Knowledge Base

ДОДАТОК Г

Відомість кваліфікаційної роботи

Позначення					Найменування		Дод. відомості	
					Текстові документи			
1.					Пояснювальна записка		56 с.	